Table of Contents

1 Introduction

1.1 Objective

The halogen iodine exists mainly as a diatomic Molecule I_2 which exhibits excellent properties for spectroscopy. The absorption band of the molecule is in the visible part of the spectrum and additionally there is only one naturally occurring stable isotope of iodine. This is of great importance, due to the fact that multiple isotopes of a molecule lead to a mixture of the absorption bands, causing difficulties in evaluation.
The first part of the experiment consists of measuring the absorption spectrum of the I_2 molecule with a CCD spectrometer. The vibrational bands pertaining to the progression $\nu'' = 0$ and the dissociation energy are to be identified from the absorption spectrum. Several other molecular constants are to be determined utilising a Birge-Sponer-Plot and the approximation with a Morse potential.
The second part of the experiment deals with the emission spectrum induced by a Helium-Neon laser. The transitions excited by the laser and their relative intensity are to be identified .

1.2 Theoretical Background

1.2.1 The Born-Oppenheimer approximation

It is impossible to analytically solve the Schrödinger equation for a molecule. The Born-Oppenheimer approximation alleviates this problem by accounting for the monumental difference in the mass of the electrons and the nuclei. Because of their small mass, electrons can react to the movement of the nuclei roughly instantaneously. The approximation is made by viewing the nuclei as stationary and solving the Schrödinger equation for the electrons (at position r_i) in the static Coulomb potential caused by the nuclei in this position (R_j). The total wave function Ψ can then be seen as being comprised of the product of the wave function of the electrons ψ and that of the nuclei χ:

$$\Psi(r_i, R_j) = \psi(r_i, R_j)\chi(R_j) \tag{1}$$

Molecular potential energy curves of a diatomic molecule can be established from the results. The Born-Oppenheimer approximation yields accurate results for ground electronic states. However for excited states it becomes less exact.

1.2.2 Molecular vibration

I. Vibrational energy levels of diatomic molecules

The potential energy of a diatomic molecule can be expressed by a Taylor series for small displacements $x = R - R_e$ from the equilibrium position:

$$V(x) = \underbrace{V(0)}_{=0} + \underbrace{\left(\frac{\mathrm{d}V}{\mathrm{d}x}\right)_0}_{=0} x + \frac{1}{2}\left(\frac{\mathrm{d}^2V}{\mathrm{d}x^2}\right)_0 x^2 + \underbrace{\frac{1}{3!}\left(\frac{\mathrm{d}^3V}{\mathrm{d}x^3}\right)_0 x^3 + ...}_{\approx 0} \tag{2}$$

As the absolute potential energy of the molecule is not of interest here, one can set $V(0) = 0$. The equilibrium corresponds to the minimum of the potential and the first derivative is zero by definition. If the displacement is very small, the third-order term becomes negligible.

This ultimately leads to the familiar result of a parabolic (harmonic) potential energy:

$$V(x) = \frac{1}{2}kx^2 \qquad\qquad k = \left(\frac{\mathrm{d}^2V}{\mathrm{d}x^2}\right)_0 \tag{3}$$

Introduction of the effective mass μ leads to the following Hamiltonian:

$$H = -\frac{\hbar^2}{2\mu}\frac{\mathrm{d}^2}{\mathrm{d}x^2} + \frac{1}{2}kx^2 \qquad \text{with} \qquad \mu = \frac{m_1 m_2}{m_1 + m_2} \tag{4}$$

The solution of the Hamiltonian with parabolic potential energy corresponds to that of the harmonic oscillator:

$$E_\nu = \left(\nu + \frac{1}{2}\right)\hbar\omega \qquad \text{with} \qquad \omega = \sqrt{\frac{k}{\mu}} \qquad \text{and} \qquad \nu \,\epsilon\, \mathbb{N} \tag{5}$$

The energy levels lie in a uniform ladder with separation $\hbar\omega$ and the wave functions are given by a product of Gaußian bell function and Hermite polynomials.

II. Anharmonic oszillation and the Morse potential

The termination of the Taylor expansion of the molecular potential energy after the quadratic term does not display the correct behaviour at large displacements from the equilibrium. In reality the energy levels converge and do not remain equidistant. This means that an *anharmonic vibration* is introduced as the vibrational excitation of the molecule is increased. A function for the molecular potential energy, which accounts for this is the *Morse potential energy*:

$$V(x) = D_e(1 - e^{-a(r-r_e)})^2 \tag{6}$$

The parameter D_e is the depth of the potential, a is a molecular constant and r_e is the internuclear distance. The repulsion at small distances is caused by the Coulomb potential. The Van der Waals force causes attraction at greater distances. Solving the Schrödinger equation with this potential analytically gives rise to the following:

$$E_\nu = \left(\nu + \frac{1}{2}\right)\hbar\omega_e - \left(\nu + \frac{1}{2}\right)^2 \hbar\omega_e x_e \qquad \text{with} \qquad \omega_e x_e = \frac{a^2\hbar}{4\pi c\mu} \qquad \omega_e = a\sqrt{\frac{\hbar D_e}{\pi c\mu}} \tag{7}$$

The quantity $\omega_e x_e$ is known as the *anharmonicity constant* and the term containing it reflects the convergence of energy levels at higher values of the vibrational quantum number ν. The number of bound states in the Morse potential is finite. The dissociation energy is given by:

$$D_e = \frac{\omega_e^2}{4\omega_e x_e} \tag{8}$$

The following figure shows a comparison of the real, parabolic and Morse potentials:

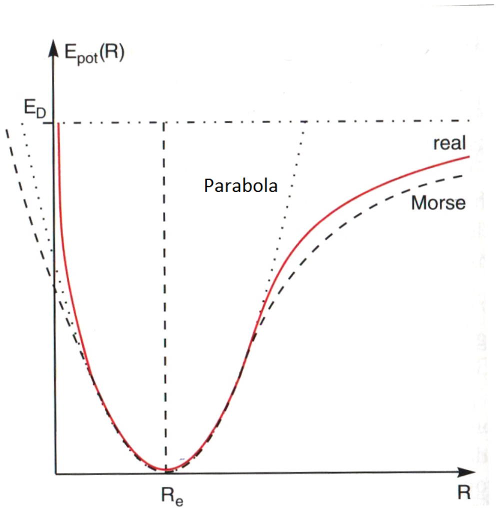


Figure 1: Plot of real, parabolic and Morse potentials (from [1])

III. Birge-Sponer-Plot

As previously mentioned the harmonic potential is an insufficient approximation at high displacements. Inserting the expanded potential energy into the Schrödinger equation results in the following energy levels after application of perturbation theory (energy given in wavenumbers):

$$G(\nu) = \omega_e \left(\nu + \frac{1}{2}\right) - \omega_e x_e \left(\nu + \frac{1}{2}\right)^2 + \omega_e y_e \left(\nu + \frac{1}{3}\right)^3 + ... \tag{9}$$

Once again ω_e is the oscillation frequency (vibrational constant) and $\omega_e x_e$, $\omega_e y_e$ are the anharmonicity constants. The difference between two adjacent energy terms is given by:

$$\Delta G\left(\nu + \frac{1}{2}\right) = G(\nu+1) - G(\nu) = \omega_e - \omega_e x_e (2\nu + 2) + \omega_e y_e \left(3\nu^2 + 6\nu + \frac{13}{4}\right) + ... \tag{10}$$

The plot of the energy differences $\Delta G(\nu + \frac{1}{2})$ against the vibrational quantum number $\nu + \frac{1}{2}$ is known as the *Birge-Sponer-Plot*. By fitting the above formula (10) to the measured data, one can obtain the aforementioned molecular constants. The fact that $\omega_e x_e \gg \omega_e y_e$ means that the energy differences will decrease with ν. Because ω_e is positive there exists a maximum vibrational quantum number ν_{diss}, which is defined by:

$$\Delta G\left(\nu_{diss} + \frac{1}{2}\right) = 0 \tag{11}$$

The value of ν_{diss} can be obtained directly from the intersection of the fitted function with the x-Axis.

The dissociation energy measured from the lowest level $\nu = 0$ is:

$$D_0 = \sum_{\nu=0}^{\nu_{diss}} \Delta G\left(\nu + \frac{1}{2}\right) \tag{12}$$

Measured from the minimum of the potential, the dissociation energy becomes:

$$D_e = G(0) + D_0 \tag{13}$$

1.2.3 Molecular rotation

Electromagnetic waves can exercise torque on a molecule and excite rotational levels. According to classical mechanics the *moment of intertia* I of two spheres (of same mass m) attached to a weightless rod of length R is:

$$I = \frac{m^2}{M}R^2 = \mu R^2 \tag{14}$$

This moment of inertia can be assigned to a diatomic homonuclear molecule (for example I_2). The kinetic energy of rotation T associated with the rotation about an axis that is perpendicular to its figure axis, is given by:

$$T = \frac{L^2}{2I} = \frac{J(J+1)\hbar^2}{2\mu R^2} = \frac{J(J+1)\hbar^2}{2I} = BJ(J+1) \qquad B = \frac{\hbar}{4\pi c I} = \frac{\hbar}{4\pi c \mu R^2} \tag{15}$$

It is convention to express the energy in terms of the *rotational constant* B.

1.2.4 Molecular electronic transitions

Absorption or emission of electromagnetic waves induce electronic transitions in molecules. As the Coulomb force changes due to the redistribution of electronic charge, the molecule begins to vibrate at a different frequency. The effect of this can be seen in the absorption spectrum. The combination of electronic and vibrational transitions is dubbed *vibronic transition*. Vibronic transitions exist from all vibronic states of the ground to those of the excited states (see 1.2.4). The transition from one specific vibronic state are grouped together to form a *progression*. Additionally to this rotational transitions also occur. These are less energetic than vibrational (and therefore electronic) transitions and are classified as *bands* perched on top of each and every vibronic state. The plethora of the different states can be seen in following graph:

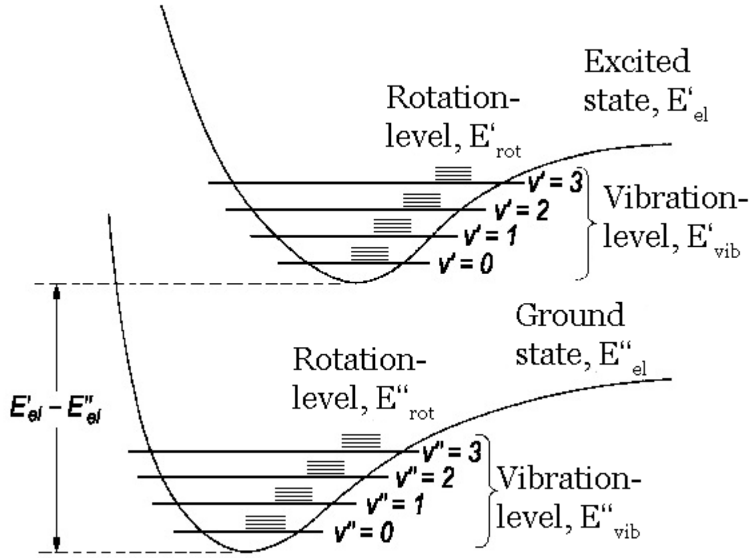


Figure 2: Structure of the different states (from [2])

The value for the energy corresponding to a specific vibronic transition (in this case emission) can be calculated theoretically with following formula (T_e is the excitation energy):

$$\Delta G = G'(\nu') - G''(\nu'') + T_e = \omega_e' \left(\nu' + \frac{1}{2}\right) - \omega_e'' \left(\nu'' + \frac{1}{2}\right) - \omega_e' x_e' \left(\nu' + \frac{1}{2}\right)^2 + \omega_e'' x_e'' \left(\nu'' + \frac{1}{2}\right)^2 + T_e \tag{16}$$

I. Spectroscopic notation

The transition of I_2 examined in the experiment is:

$$\mathrm{X}^1\Sigma_{0g}^+ \leftrightarrow \mathrm{B}^3\Pi_{0u}^+ \tag{17}$$

The X indicates the ground- and the B the second excited state. The upper case number to the left of the capital Greek letter is the *multiplicity* of the molecule. The multiplicity corresponds to $2S + 1$, S being the total spin of the electron configuration. A system of multiplicity 1 is known as *singlet* and of 3 as *triplet*.

The capital Greek letters are the total orbital angular momentum around the internuclear axis (referred to as Λ). The corresponding Greek letters to the values of Λ are:

$$\Lambda = 0, 1, 2, 3, .. \qquad \longleftrightarrow \qquad \Sigma, \Pi, \Delta, \Phi, ... \tag{18}$$

Λ is calculated by summing up the individual orbital angular momentum values (λ). These are obtained from the LCAO[1] method.
The superscript + or − to the right of the Greek letter gives information about the effect of reflection in plane containing the internuclear axis.
The lower case g or u to the right of the Greek letter indicate the *parity* of the molecular orbitals. This is governed by their symmetry properties under inversion of the electron coordinates. If the molecular orbitals remain unfazed under inversion they are denoted by a g (German for "even"). If however the inversion causes a sign change, they are denoted by u (German for "odd").

A further characteristic parameter for diatomic molecules arises from the *Hund coupling cases.* The total angular momentum of the molecule has a component $\Omega\hbar$ parallel to the internuclear axis. Ω is defined as the sum of Λ and components of spin angular momenta on the internuclear axis:

$$\Omega = \Lambda + \Sigma \tag{19}$$

II. Selection rules

The electronic transitions of diatomic molecules are subject to the following *selection rules*:

- g $\leftrightarrow$ u, g $\not\leftrightarrow$ g, u $\not\leftrightarrow$ u
- $\Sigma^{+/-} \leftrightarrow \Sigma^{+/-}$, $\Sigma^{+/-} \not\leftrightarrow \Sigma^{-/+}$
- $\Delta\Lambda = 0, \pm 1$
- $\Delta\Omega = 0, \pm 1$
- ΔS= 0

The observed transition (17) seems to violate the last selection rule. Usually *"spin flips"* are forbidden, but because iodine is a rather heavy atom *spin-orbit coupling* sets in. This acts as a perturbation that mixes states of different multiplicity.

[1] Linear combination of atomic orbitals

III. Franck-Condon principle

As the mass of an electrons is far smaller than that of the nuclei, the nuclear positions during an electronic transition stay the same. They adjust once the electrons stop moving. This can be seen in the following figure:

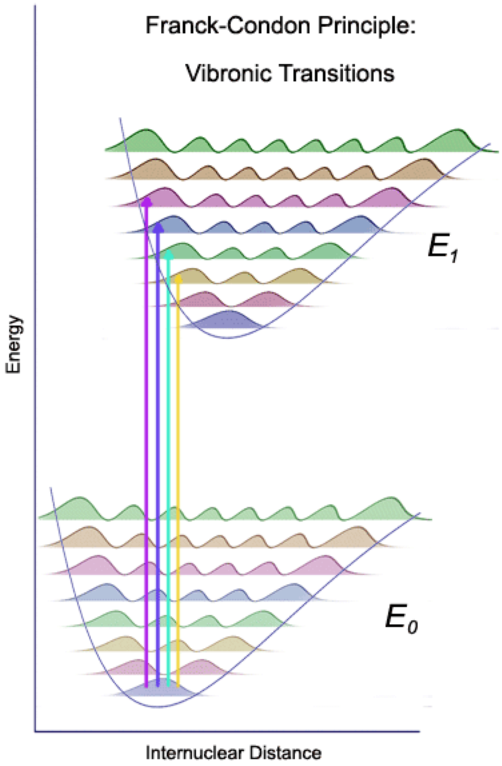


Figure 3: Quantum mechanical version of the Franck-Condon principle

The upper potential curve is shifted along the x (Internuclear distance)-axis, because excitation leads to higher antibonding. The transitions from one state happen, that are most similar to the end state. The wavefunction that bears the highest resemblance to the original is given by the one with a peak directly above the ground state. The transitions are therefore called *vertical*. The absolute square of the overlap integral of two electronic states is called the *Franck-Condon factor*:

$$FC(\nu', \nu'') = \left| \int \psi'_\nu(R) \psi''_\nu(R) \mathrm{dr} \right|^2 \tag{20}$$

The Franck-Condon factor defines the relative intensity of the electronic transition and can be calculated by inserting the wavefunctions, given by the product of Gaussian bell functions and Hermite polynomials.

2 Experimental Procedure

2.1 Experimental set-up

I. Absorption spectrum

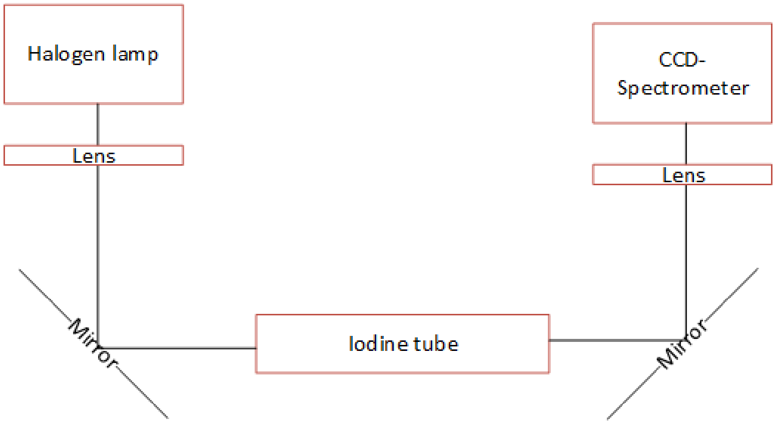


Figure 4: Experimental set-up for the absorption spectrum

Light emitted from a halogen lamp collimated by a lens and channelled into the iodine tube after reflection by a mirror. The light leaving the iodine tube is once again reflected by a mirror and then focused onto a CCD-spectrometer by another lens. The CCD-spectrometer collects data and transfers it to the computer nearby.

II. Emission spectrum

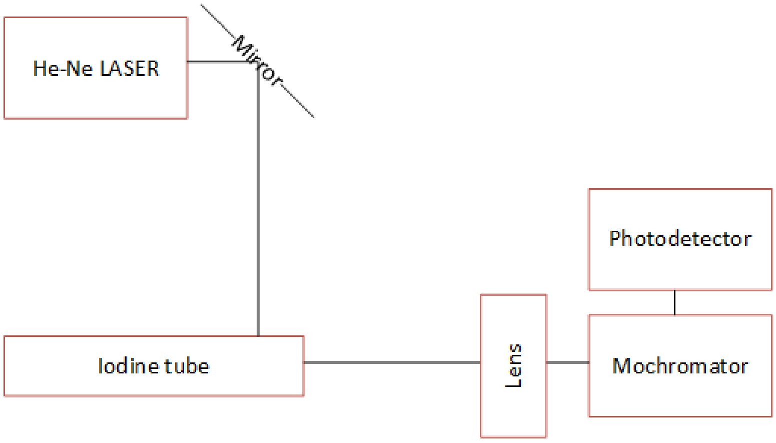


Figure 5: Experimental set-up for the emission spectrum

For the absorption spectrum the second mirror of the previous set-up is replaced with the second lense. The light now enters a monochromator and then a photomultiplier which sends data the computer using a different measuring programme as before.
Before the emission spectrum is examined the monochromator is calibrated by swapping the halogen- with a mercury-vapour lamp and measuring the resulting discrete spectrum. The comparison of the

measured wavelengths of the spectral lines with the expected ones can be used to correct an offset that the monochromator might exhibit.
The light required for the obtaining of an emission spectrum originates from a Helium-Neon laser. The laser light is guided into the iodine by a mirror on the top side of the tube. The radiation leaving the tube is captured by the monochromator and the photomultiplier. The iodine has to be present in gas form. Despite its high boiling temperature, a fair amount of iodine is gaseous at room temperature. To ensure a sufficient volume of gaseous iodine, it is advised to make liberal use of the freely available hair dryer as a constant heat source.

2.2 Tasks

I. Absorption

1. Record the absorption spectrum of I_2 with the CCD-spectrometer.
2. Identify the vibrational bands of the progression $\nu'' = 0$
3. Calculate the constants ω_e' and $\omega_e' x_e'$ using the Birge-Sponer-Plot.
4. Calculate the dissociation energy D_e in two different ways:
   a) Using the approximation of the Morse potential
   b) Summing over the energy differences
5. Determine the energy E_{diss} under which the I_2 molecule dissociates, from the absorption spectrum.
6. Sketch the Morse potential for the excited state.

II. Emission

1. Record the spectrum of the mercury vapour lamp for wavelengths of $400-600$ nm for calibration purposes.
2. Record the emission spectrum of iodine of the I_2 molecule for wavelengths of $600 - 800$ nm.
3. Carry out the calibration of the monochromator.
4. Identify the transition excited by the laser.
5. The relative intensities of the Stokes lines are to be examined.

3 Evaluation

The entire evaluation was performed with *R*. The literature values were taken from the papers [5], [6] and [7].

3.1 Absorption

3.1.1 Spectrum of the halogen lamp

The spectrum of the halogen lamp was measured without the iodine tube. As expected the spectrum spans the entire visible wavelengths and has no absorption lines:

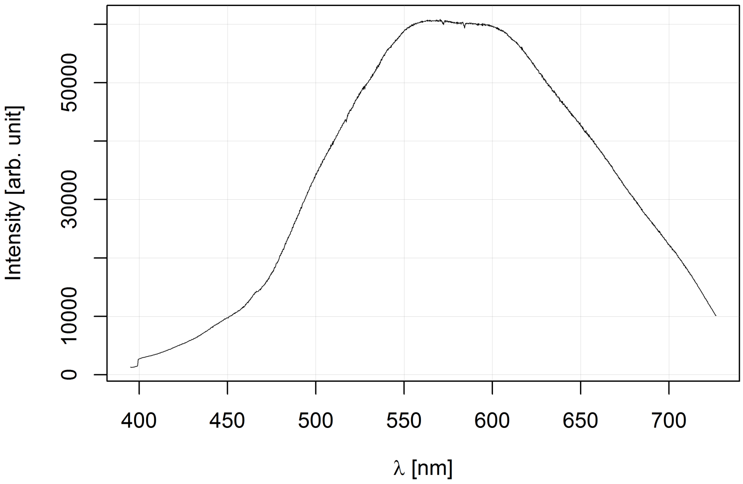


Figure 6: Spectrum of the halogen lamp

3.1.2 Identification of the vibrational bands pertainig to $\nu'' = 0$

The absorption spectrum was able to be recorded with the CCD-spectrometer:

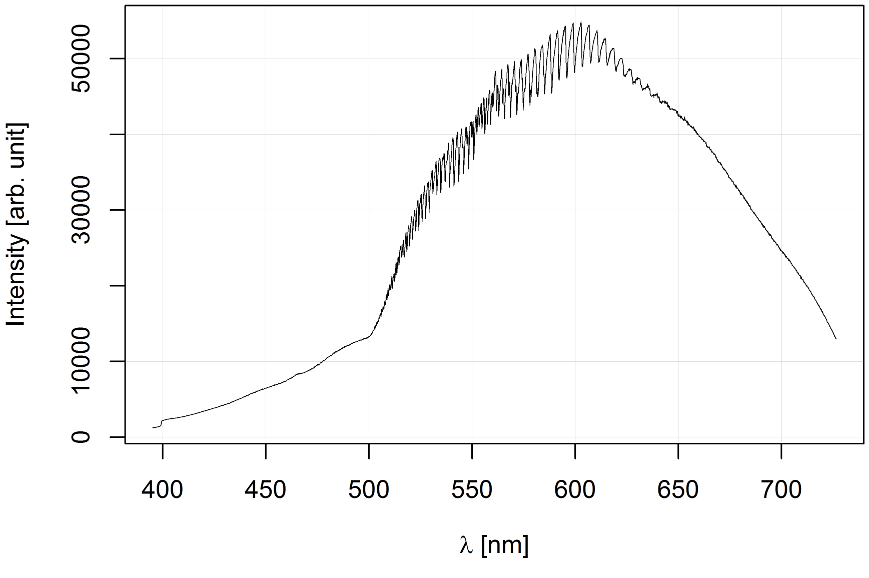


Figure 7: Overview of the Absorption Spectrum

The identification of the bands is made easier by enlarging the segment of ca. 500-580 nm and using the knowledge that the absorption band at a wavelength of $\lambda = 545,8\,\mathrm{nm}$ stems from the transition $\nu'' = 0 \rightarrow \nu' = 25$. The 25th band of the progression is marked by a vertical green line in the plot. The distances between the dips of one progression become smaller with lower wavelengths because they correspond to the difference in energy between the vibrational levels in the excited potential. These converge to zero as discussed above.

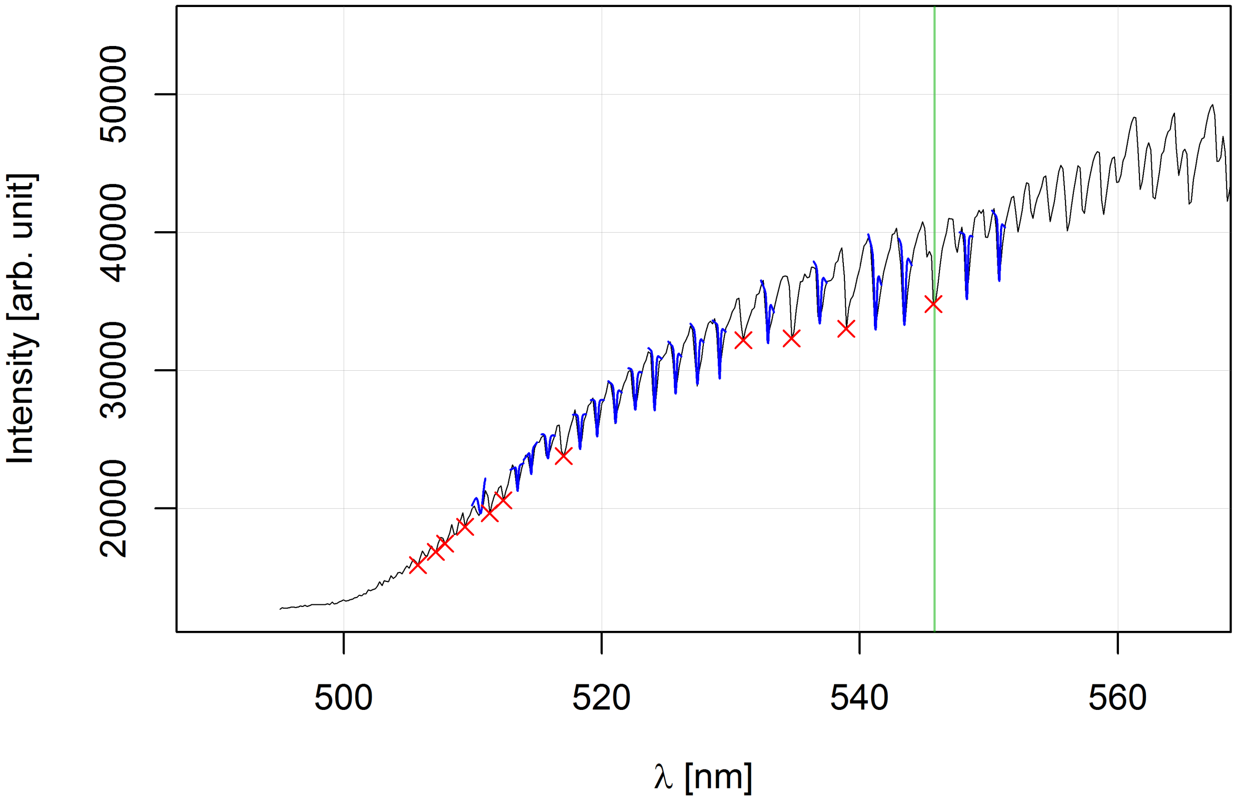


Figure 8: Enlarged of the Absorption Spectrum

The dips associated with the $\nu'' = 0$ band are marked in blue or with red crosses. Looking at lower values of the wavelength one is able to identify 27 further bands (up to $\nu'' = 0 \rightarrow \nu' = 52$). The identification of more bands at lower wavelengths becomes very difficult as the dips shrink rapidly. Two additional bands were identified to the right of the given 25th band (down to $\nu'' = 0 \rightarrow \nu' = 23$). The presence of dips caused by the first (or even second) progressions at higher wavelengths makes the classification of more bands very tedious and prone to errors. An attempt was made to fit Gauss-functions to the peaks confined by the absorption dips. The fits that resulted in success are displayed in blue. The dips for which the fit did not converge are marked with red crosses. The uncertainty of the wavelength of the absorbed radiation can be extracted from the standard deviation of these fits. The uncertainty was set as half the standard deviation of the Gaussian functions (reasoning on following page). The uncertainty of neighbouring dips was used for the ones for which the fits failed.

3.1.3 Calculation of ω_e' and $\omega_e' x_e'$ from the Birge-Sponer plot

The Birge-Sponer plot for the zero progression is obtained by plotting the energy differences $\Delta G\left(\nu' + \frac{1}{2}\right)$ against the vibrational quantum number $\tilde{\nu} = \nu' + \frac{1}{2}$.

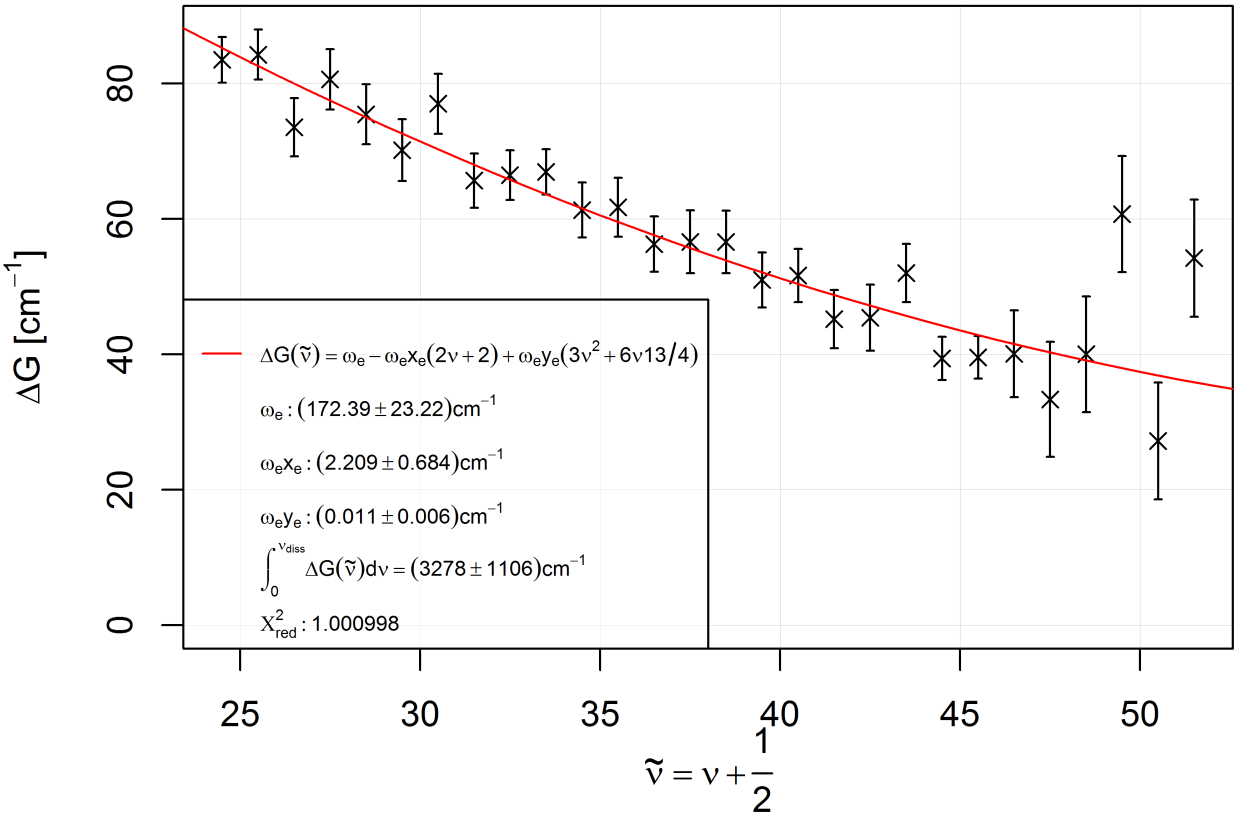


Figure 9: Birge-Sponer plot with quadratic fit

The energy differences are in the unit of wavenumbers:

$$\Delta G\left(\nu' + \frac{1}{2}\right) = \frac{1}{\lambda(\nu' + 1)} - \frac{1}{\lambda(\nu')} \tag{21}$$

The error bars originate from the standard deviation of the Gauss-fits, as described above. The uncertainties of the energy differences (the error bars) are given by error propagation:

$$s_{\Delta G(\nu' + \frac{1}{2})} = \sqrt{\left(\frac{s_{\lambda(\nu'+1)}}{\lambda(\nu' + 1)^2}\right)^2 + \left(\frac{s_{\lambda(\nu')}}{\lambda(\nu')^2}\right)^2} \tag{22}$$

The value for χ^2 (very close to 1) indicates the choice for the uncertainty of the wavelength was well founded. The molecular constants were obtained by fitting the function given by formula (**10**) to the data (see Figure 8):

$$\begin{aligned} \omega_e' &= (170 \pm 20)\,\mathrm{cm}^{-1} \\ \omega_e' x_e' &= (2 \pm 0.7)\,\mathrm{cm}^{-1} \\ \omega_e' y_e' &= (0.011 \pm 0.006)\,\mathrm{cm}^{-1} \end{aligned} \tag{23}$$

The apparent value for $\omega_e' y_e'$ obtained from the fit has an alarmingly high uncertainty (a relative error of over 50%). Because of this and the fact that the slope of the curve becomes saturated, which results in a highly unlikely intersection with the x-axis, a fit without the quadratic term was performed. The values extracted from this were used for further calculations:

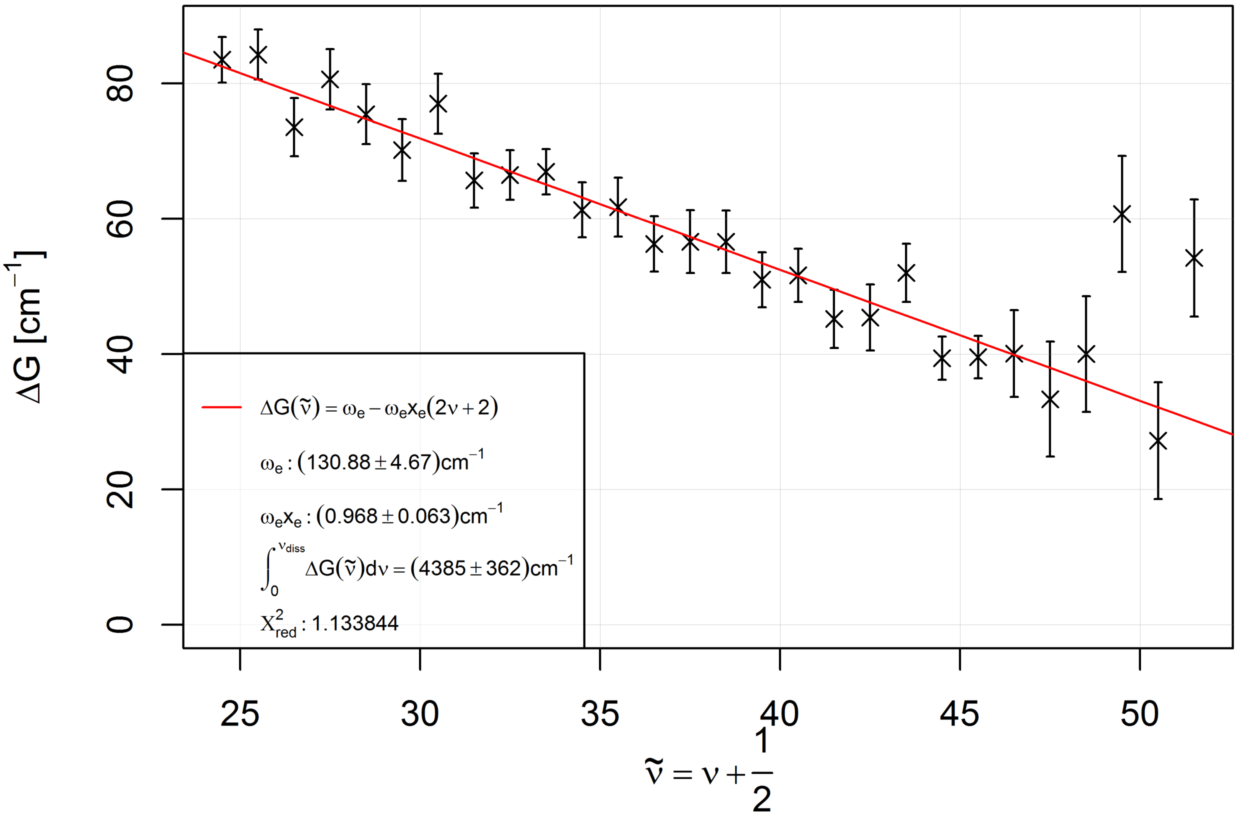


Figure 10: Birge-Sponer plot with linear fit

The results of this fit are:

$$\begin{aligned} \omega_e' &= \mathbf{(131 \pm 5)\,cm^{-1}} \\ \omega_e' x_e' &= \mathbf{(0.97 \pm 0.06)\,cm^{-1}} \end{aligned} \tag{24}$$

The literature values for the molecular constants are:

$$\begin{aligned} \omega'_{e_{\text{lit}}} &= \mathbf{125.273\,cm^{-1}} \\ \omega_e' x'_{e_{\text{lit}}} &= \mathbf{0.70163\,cm^{-1}} \end{aligned} \tag{25}$$

The experimental value for ω_e is compatible with the literature value in a range of 1σ. In contrast the value for $\omega_e' x_e'$ shows a discrepancy of around about 4σ. This could be explained by an insufficient amount of data, particularly at higher vibrational quantum numbers.

3.1.4 Calculation of the dissociation energy of the B$^3\Pi_{0u}^+$ state

The dissociation energy D_e of the excited state is determined with two different methods. The literature value is:

$$D_{e_{lit}} = 4391\,\mathrm{cm}^{-1} \tag{26}$$

a) Approximation with Morse potential

The dissociation energy can be calculated using formula (**8**), with following uncertainty:

$$D_e = \frac{\omega_e^2}{4\omega_e x_e} \qquad\qquad s_{D_e} = \sqrt{\left(\frac{2\omega_e}{4\omega_e x_e}\right)^2 s_{\omega_e}^2 + \left(\frac{\omega_e^2}{4(\omega_e x_e)^2}\right)^2 s_{\omega_e x_e}^2} \tag{27}$$

The calculated value is:

$$D_{e_{Mo}} = \mathbf{(4400 \pm 400)\,cm^{-1}} \tag{28}$$

This value boasts a rather high uncertainty and is therefore compatible with the literature value in a range of under 1σ.

b) Summing over the energy differences

As mentioned in the theoretical part the sum of all of the energy differences in the Birge-Sponer plot (from $\nu = 0$ to ν_{diss}) corresponds to the dissociation energy. However summing directly proves to be difficult, due to the lack of data for $\nu' < 23$ and $\nu' > 52$. Thankfully an alternative to the summation exists.
The area enclosed by the curve and the x-axis, which corresponds to dissociation energy, can be attained by computing following integral:

$$D_e = \int_0^{\nu_{diss}} \Delta G(\tilde{\nu})\mathrm{d}\nu' \tag{29}$$

The integer value of ν_{diss} can be found by calculating the intersection of the linear function seen in the Birge-Sponer plot with the x-axis.

$$\nu_{diss} = \frac{\omega_e}{2\omega_e x_e} - 1 \qquad \text{with} \qquad s_{\nu_{diss}} = \sqrt{\left(\frac{s_{\omega_e}}{2\omega_e x_e}\right)^2 + \left(\frac{\omega_e s_{\omega_e x_e}}{2(\omega_e x_e)^2}\right)^2} \tag{30}$$

The value for ν_{diss} obtained by evaluating these expressions is:

$$\nu_{diss} = 67 \pm 5 \tag{31}$$

The area under the curve corresponds to that of a triangle with side lengths ω_e and ν_{diss}:

$$D_e = \frac{1}{2}\omega_e \nu_{diss} \tag{32}$$

The uncertainty for the area under the curve and therefore D_e is given by:

$$s_{D_e} = \sqrt{\left(\frac{s_{\omega_e}}{\omega_e}\right)^2 + \left(\frac{s_{\nu_{diss}}}{\nu_{diss}}\right)^2} D_e \tag{33}$$

The end result for D_e using this method is:

$$D_{e_{Sum}} = (\mathbf{4400} \pm \mathbf{400})\,\mathbf{cm}^{-1} \tag{34}$$

The value once again matches the literature value in a range of 1σ.

3.1.5 Graphical determination of E$_{diss}$

The energy under which the iodine molecule dissociates E_{diss} can be extracted from the absorption spectrum by analysing the low wavelength part. The aim is to find the value of the lowest wavelength under which absorption dips are discernible. Below this limit absorption is continuous because the I_2 molecule will have dissociated into fragments which have varying kinetic energies. An enlarged image of the low wavelength part of the spectrum can be seen here:

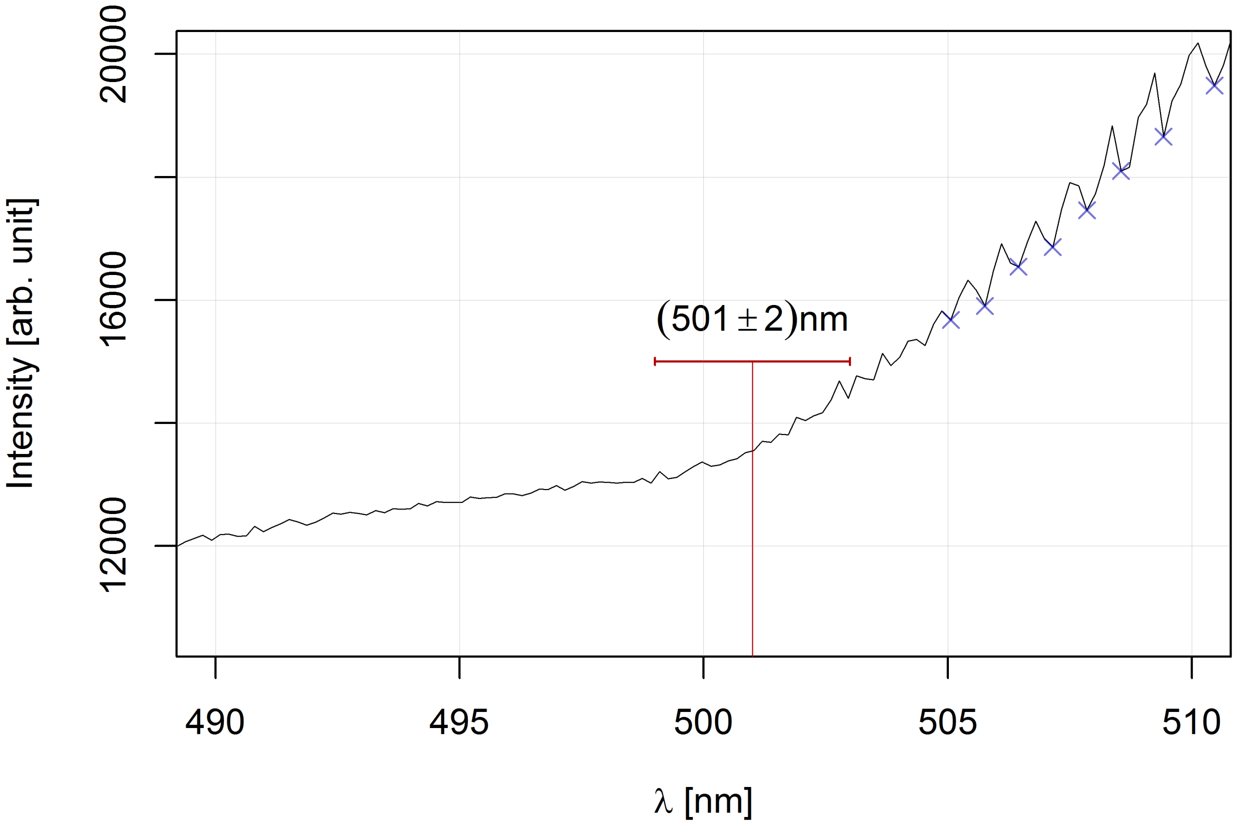


Figure 11: Enlarged low wavelength section of the absorption spectrum

The wavelength at which the dissociation commences and its uncertainty were estimated to be:

$$\lambda_{diss} = (501 \pm 2)\,\mathrm{nm} \tag{35}$$

The corresponding energy in wavenumbers is:

$$\mathbf{E_{diss} = (19960 \pm 80)\,cm^{-1}} \tag{36}$$

The literature value for the dissociation is:

$$\mathbf{E_{diss_{lit}} = 20014\,cm^{-1}} \tag{37}$$

The values are compatible in a range of 1σ.

3.1.6 Morse potential for the excited state

In order to plot the Morse potential of the excited, the constants D_e, a and r_e are required. As two values for D_e were determined in the last section, a Morse potential for each of them will be plotted. The value for a can be ascertained from the results of the measurements with the following calculation (used constants in footnote):

$$a = 2\sqrt{\frac{\omega_e x_e \pi c \mu}{\hbar}} \qquad s_a = \frac{a}{2\omega_e x_e} s_{\omega_e x_e}$$
$$a = (1.91 \pm 0.06)\frac{1}{\text{Å}} \tag{38}$$

The theoretical value for the rotation constant is $B'_e = 0.029\,\text{cm}^{-1}$. From this, one is able to calculate the bond length r_e:

$$r_e = \sqrt{\frac{\hbar}{4\pi c \mu B'_e}} = 3.026\,\text{Å} \tag{39}$$

The resulting Morse potentials can be seen below:

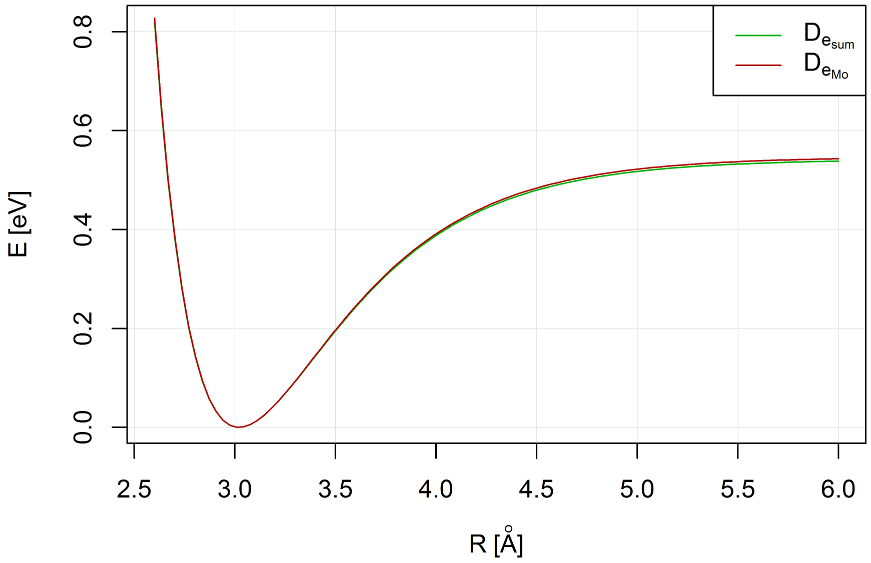


Figure 12: Morse potential of the excited state

$\mu = 1.0544 \cdot 10^{-25}\,\text{kg}$
$c = 3 \cdot 10^{8}\,\frac{\text{m}}{\text{s}}$
$\hbar = 1.0546 \cdot 10^{-34}\,\text{Js}$

3.2 Emission

3.2.1 Calibration with mercury vapour lamp

The result of the calibration measurement with the mercury vapour lamp can be seen here:

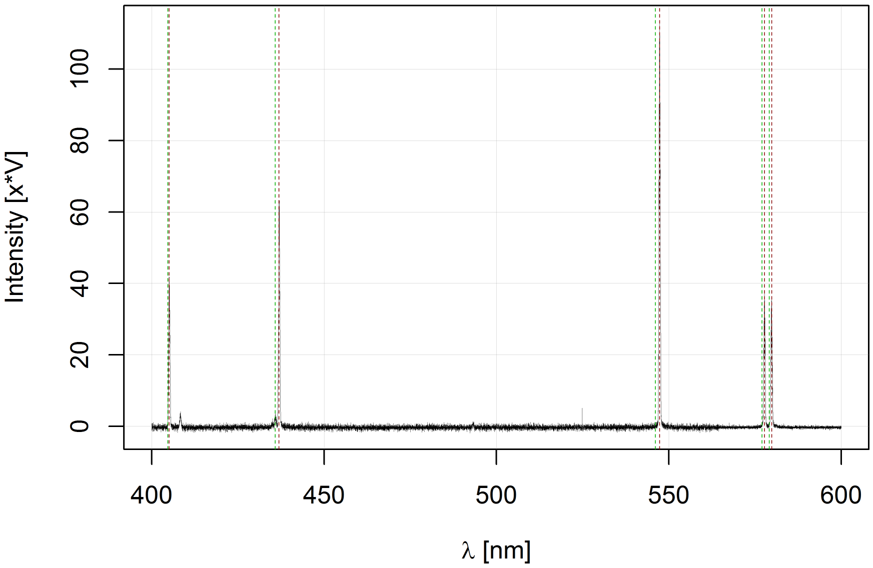


Figure 13: Measured mercury vapour lamp

The wavelengths displayed by the computer must be converted into their true wavelength. This is accomplished by comparing the wavelengths of the measured peaks with the literature values and taking the mean ($\overline{\Delta\lambda}$) of the differences:

Literature value / nm	Measured wavelength / nm	$\Delta\lambda$
404.66	405.06	-0.4
435.83	436.92	-1.09
546.07	547.3	-1.23
576.96	577.73	-0.77
579.07	579.8	-0.73

$$\Rightarrow \qquad \overline{\Delta\lambda} = -0.84$$

Table 1: Literature and experimental values for the spectral lines

The mean value was subsequently subtracted from the measured wavelength in order to obtain the corrected wavelength.

3.2.2 Laser light peak

The mirror and lens of the experimental set-up for the emission spectrum was adjusted in order to ensure maximum intensity for the emission peaks. Afterwards the peak originating from transmitted laser light was recorded:

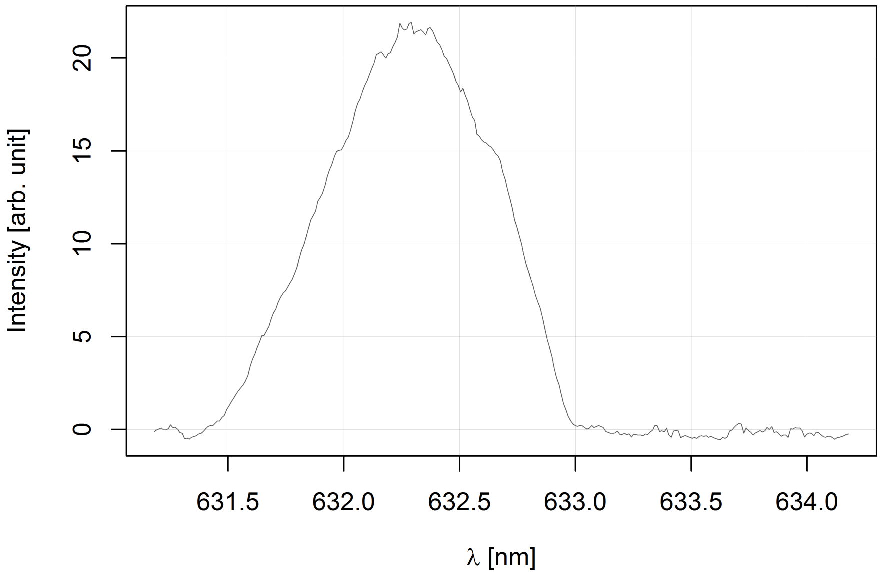


Figure 14: He-Ne laser peak (slightly smoothed)

The wavelength at the maximum of the peak was determined to be $\lambda_{max} = (632.3 \pm 0.9)\,\text{nm}$. The uncertainty for the peak maximum was estimated by fitting a Gaussian function to the peak (see fig. below) and taking the FWHM (Full width at half maximum). The expected value for the maximum is $\lambda_{He-Ne} = 632.8\,\text{nm}$. These values are compatible in a range of 1σ. This indicates that the calibration contained no serious error.

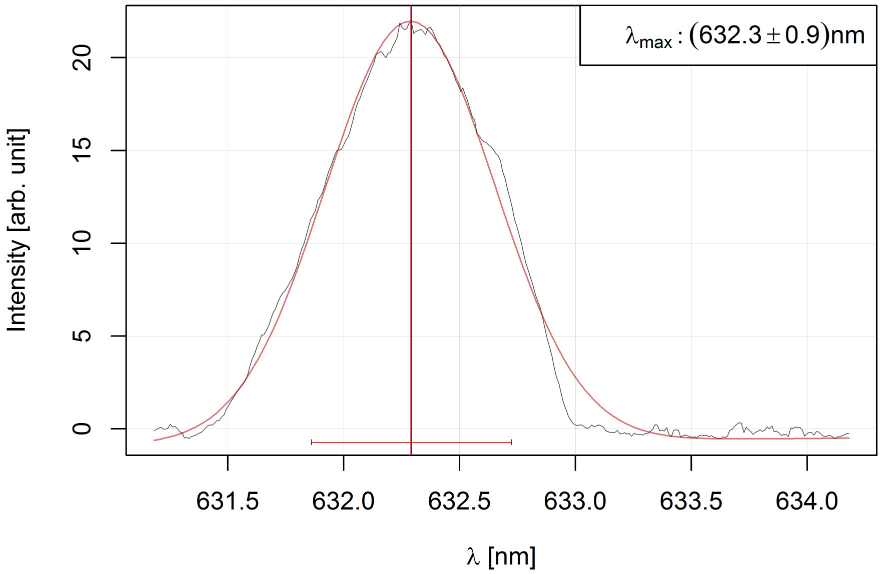


Figure 15: He-Ne laser peak with Gaussian fit

3.2.3 Emission spectrum of I_2

The recorded emission spectrum with marked emission lines can be seen below. The y-axis is given in relative intensity (relative to the intensity of the highest peak)

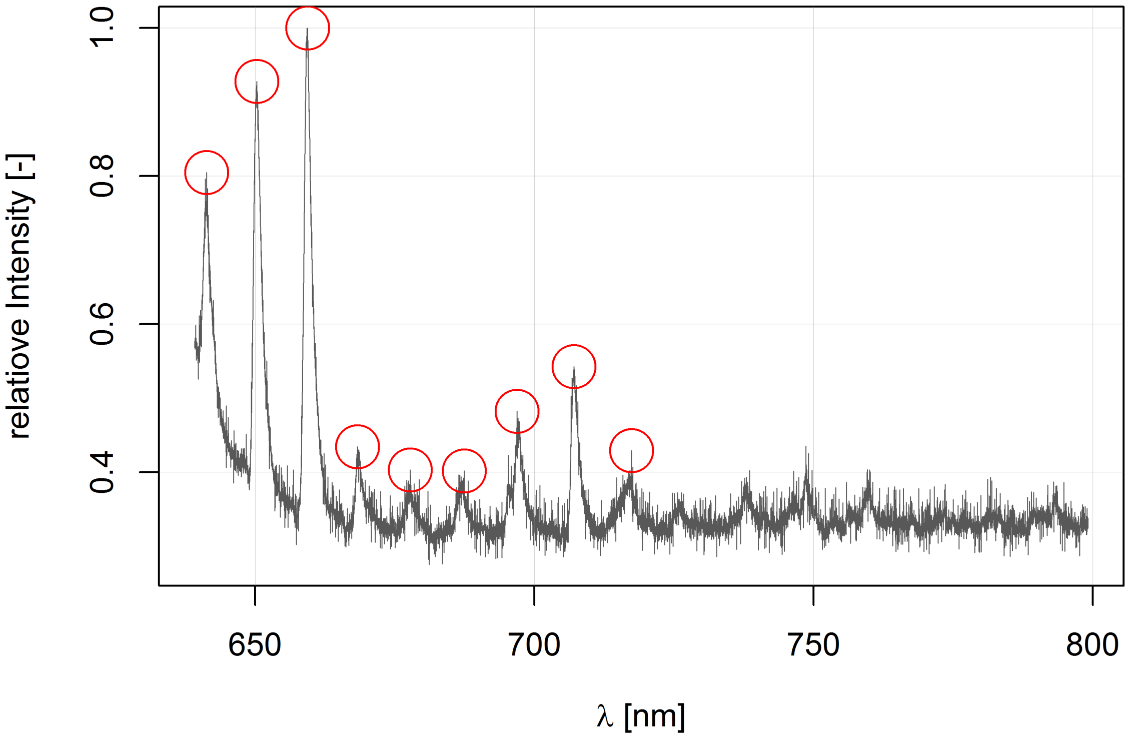


Figure 16: Measured emission spectrum

The wavelengths at which emission occurs could be taken straight from the graph. The error was determined by fitting Gaussian functions to the peaks and using their FWHM as uncertainty. The wavelengths and corresponding energy of the peaks are listed in the table below:

Peak number	λ / nm	G / cm^{-}1
1	$(\mathbf{641.3 \pm 1.4})$	$(\mathbf{15590 \pm 70})$
2	$(\mathbf{650.3 \pm 1.3})$	$(\mathbf{15380 \pm 60})$
3	$(\mathbf{659.4 \pm 1.3})$	$(\mathbf{15170 \pm 60})$
4	$(\mathbf{668.3 \pm 1.1})$	$(\mathbf{14960 \pm 50})$
5	$(\mathbf{677.8 \pm 1.9})$	$(\mathbf{14760 \pm 80})$
6	(687.4 ± 2.0)	(14550 ± 90)
7	(696.9 ± 1.2)	(14350 ± 50)
8	(707.1 ± 1.3)	(14140 ± 50)
9	(717.4 ± 1.6)	(13940 ± 60)

Table 2: Experiment values for the wavelengths and energies of the emission lines

The vibronic transitions of the Stokes lines in the measured emission spectrum can be identified by comparison with the theoretically calculated values for the energy (using formule (16)). This was done for the first five maxima. The assigned energies

ν'	$G(\nu''=4)$	$G(\nu''=5)$	$G(\nu''=6)/$	$G(\nu''=7)/$	$G(\nu''=8)$	$G(\nu''=9)$
1	15004	14795	14588	14382	14177	13974
2	15126	14918	14711	14505	14300	14096
3	15247	15039	14832	14626	14421	14217
4	15367	15159	14951	14745	14541	14337
5	15485	15277	15070	14864	14659	14455
6	**15602**	**15394**	**15187**	**14980**	**14776**	14572
7	15718	15509	15302	15096	14891	14688
8	15832	15623	15416	15210	15005	14802
9	15944	15736	15529	15323	15118	14914

Table 3: Theoretical energy values for the transitions (in wavenumbers)

The Stokes lines of the emission spectrum originate from the $\nu' = 6$ and $\nu' = 11$ excited levels ([7]). The first five maxima match the energies of the $\nu' = 6$ excited state within their respective errors. They were assigned the following transitions:

Peak	Transition
1	$\nu' = 6 \to \nu'' = 4$
2	$\nu' = 6 \to \nu'' = 5$
3	$\nu' = 6 \to \nu'' = 6$
4	$\nu' = 6 \to \nu'' = 7$
5	$\nu' = 6 \to \nu'' = 8$

Table 4: Assigned transition

The relative intensities of each peak are:

Peak	Relative intensity
1	0.805
2	0.928
3	1.000
4	0.434
5	0.403

Table 5: Measured relative intensities

The higher the relative intensity of the peak, the higher the probability for this transition. The resulting fluorescence spectrum can be compared to the experimental results from the paper by *Palmer, Cruickshank and Lewis* [7]. The emission spectrum that they were able to record can be seen below:

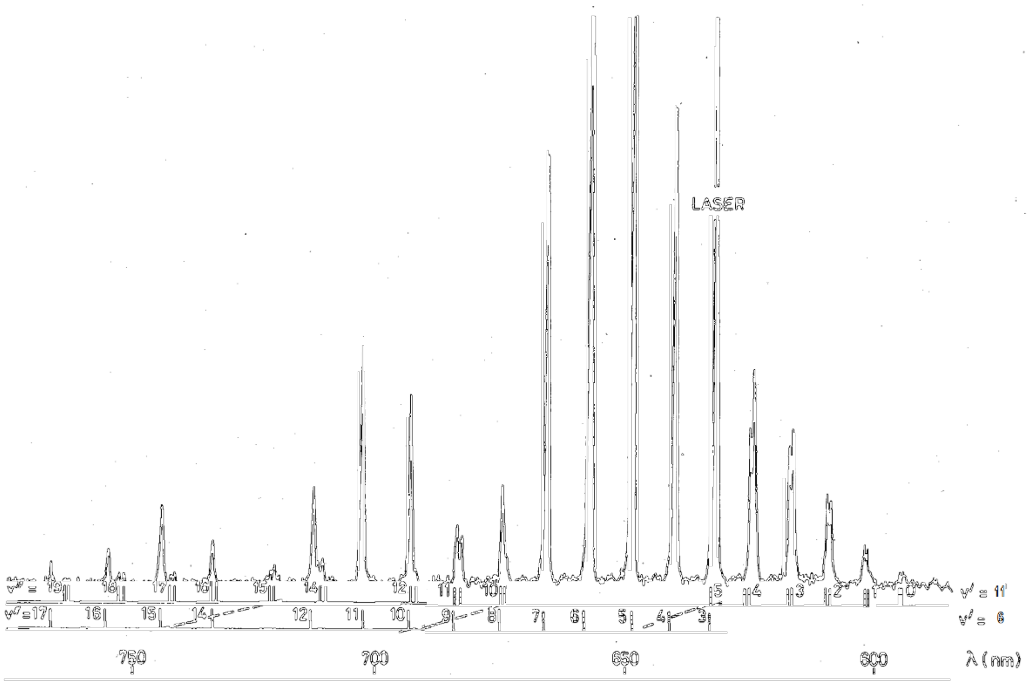


Figure 17: Experimental results from [7] for comparison

The positions of the peaks are very similar in both emission spectra. The relative intensities of the Stokes lines are also roughly the same. The only exception is the fourth maximum in the measured spectrum, which has a relative intensity of only 0.434. Judging from the experimental results of the publication this should be around about the relative intensity of the first peak. In the experiment the first peak is nearly twice as high as the fourth one. This can be explained by slight alterations of the beam path during the measurement and irregular heating of the iodine tube. This might have resulted in an inhomogeneous distribution of the gaseous iodine. Furthermore the presence of an absorbing substance of exactly this wavelength may explain the drastic decline of the height of the peak.

All in all the recorded emission spectrum appears to offer rather good qualitative information on the fluorescence induced by a Helium-Neon laser. Although there seems to be considerable amount of background noise afflicting the measurements.

4 Summary

4.1 Absorption

Following values for the molecular constants were obtained by using a Birge-Sponer plot:

	Measured value / cm^{-1}	Literature value / cm^{-1}
ω_e'	(131 ± 5)	125.273
$\omega_e' x_e'$	(0.97 ± 0.06)	0.70163

Table 6: Literature and experimental values for the vibrational and anhrmonicity constant

The values for the vibrational constants match in a range of 1σ. However the anharmonicity constants are 4σ apart.
The main reason for this is that the quadratic function which was fitted to the data was approximated linearly. The identification of additional bands at lower wavelengths would have drastically improved the end result, as the function is quadratic in the vibrational quantum number.
The limited spectral resolution of the CCD-spectrometer did not suffice for an exact identification of the absorption dips. This may have been exacerbated by the presence of noise caused by dark current inherent to the CCD-spectrometer.
The lamp and the correct set-up of the beam path did not prove to be very problematic in the measuring of the absorption spectrum.

Using the molecular constants extracted from the Birge-Sponer plot the dissociation energy of the excited ($B^3\Pi_{0u}^+$) state could be determined with two methods. The first method was with the approximation of the Morse potential. The second method was summing over the energy differences in the Birge-Sponer plot. The results and the literature value can be seen in following table:

$D_{e_{Mo}}$ / cm^{-1}	$D_{e_{Sum}}$ / cm^{-1}	$D_{e_{lit}}$
(4400 ± 400)	(4400 ± 400)	4391

Table 7: Literature and experimental values for the dissociation energy of the $B^3\Pi_{0u}^+$ state

The resulting values of both methods match the literature value in a range of 1σ. Neither of the methods proved to be particularly exact. This is due to the fact that both methods incorporate the errors pertaining to the Birge-Sponer plot. Accordingly the same systematic and statistical errors apply. The approximation through the Morse potential leads to a surprisingly good result, considering its simplicity.

The dissociation energy of the iodine molecule was found graphically:

E_{diss} / cm^{-1}	Literature value / cm^{-1}
(199960 ± 80)	20014

Table 8: Literature and experimental values for the dissociation energy of the I_2 molecule

The values are compatible in a range of 1σ. Although finding the wavelength at which the discrete absorption spectrum becomes indistinguishable from a continuum was error-prone. The uncertainty was chosen accordingly. Once again the difficulty could be reduced by using a more high resolution spectrometer.

A Morse potential of the excited state was plotted with both values for the dissociation energy. The difference between the two graphs is minuscule and both the plots appear to be realistic.

In conclusion the analysis of the absorption spectrum with the Birge-Sponer plot gave rise to realistic results, despite their slightly high uncertainty.

4.2 Emission

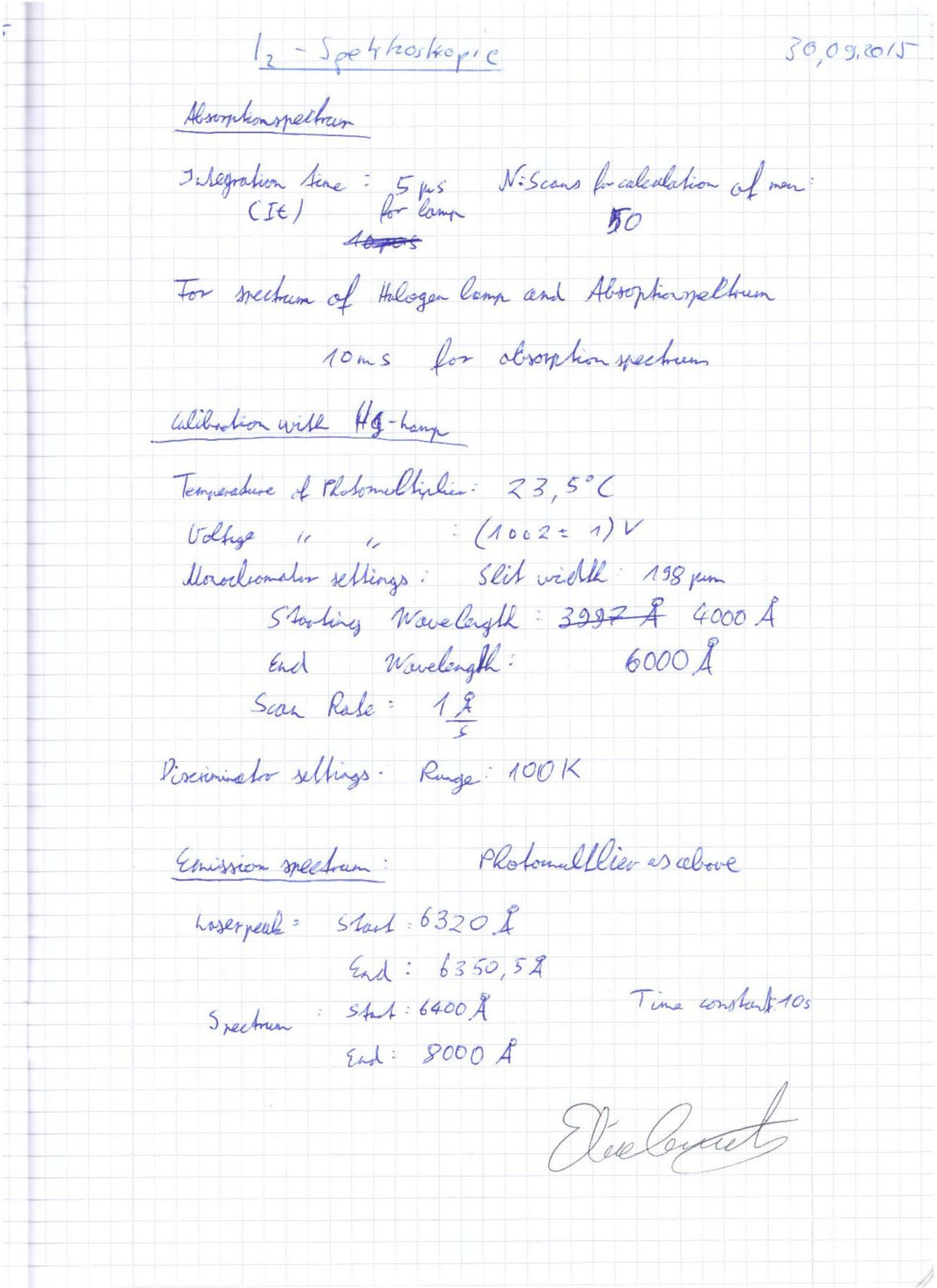
In the recorded emission spectrum the Stokes lines caused by transitions from the $\nu' = 6$ state could be identified. The relative intensities of the majority of the peaks was strikingly similar to the results of a Helium-Neon laser fluorescence experiment published in the *American Journal of Physics* in 1994. The only exception is the fourth peak. The peak is approximately half as high as expected.
A reason for this could, be as previously stated, an absorbing substance. Relying on the hair dryer as a heat source, to ensure gaseous iodine, may have resulted in an inhomogeneous distribution of it in the iodine tube. Furthermore the experimental set-up was highly sensitive to slight alterations in the change of the lens and mirror.
The monochromator used for the measurement was very old and so was the used software. The conversion of the number of steps recorded by the software into wavelengths probably led to a further systematic error which was considered in the error estimation. The estimation of the uncertainty of the peak location with the FWHM of the fitted Gaussian functions may have even been to optimistic.

A final source of error was the photomultiplier and accompanying electronics. The noise afflicting the recorded emission spectrum points to dark current in the photomultiplier or the presence of unwanted background light. As the background light was kept to a minimum the former is more likely. It is highly improbable that the photomultiplier was able to deliver constant levels of amplifications for every incident photon. Thermal noise or electronic noise could have been amplified by the photomultiplier causing the signal to deteriorate. The observed movement of the intensity indicating needle while adjusting the measuring range of the monochromator, support the concerns about the quality of the experimental apparatus.

All in all the obtained fluorescence spectrum proved to be an adequate result, providing useful information on vibronic transitions.

5 Appendix

5.1 Measured data



I_2 - Spektroskopie 30,09,2015

Absorptionspektrum

Integration time : 5 μs N: Scans for calculation of mean:
(IE) for lamp 50
~~40 ms~~

For spectrum of Halogen lamp and Absorptionspektrum

10 ms for absorption spectrum

Calibration with Hg-lamp

Temperature of Photomultiplier: 23,5°C

Voltage " " : (1002 ± 1) V

Monochromator settings : Slit width : 198 μm

Starting Wavelength : ~~3997 Å~~ 4000 Å

End Wavelength : 6000 Å

Scan Rate : $1 \frac{Å}{s}$

Discriminator settings : Range : 100 K

Emission spectrum : Photomultiplier as above

Laserpeak : Start : 6320 Å

End : 6350,5 Å

Spectrum : Start : 6400 Å Time constant: 10s

End : 8000 Å

Figure 18: Laboratory journal entry

5.2 R Source code

5.2.1 Absorption

```
graphics.off()
path = dirname(dirname(sys.frame(1)$ofile))

files = dir(paste(path, "Absorption", sep = "/"), full.names = TRUE, pattern = ".txt")
#filter Files
# files = files[grepl("Jod2", files)]

# some constants
h  = 6.6026e-34
hbar = h / 2 / pi
c_  = 3e8
q  = 1.602177e-19

# make filename
for(file_ in files[grepl("keinJod", files)]){
        cat("\nstarting ", basename(file_), sep ="")
        flush.console()
        # read file to get header and relevant linenumbers
        temp_ = readLines(file_)
        skip = grep(">>>>>Begin Processed Spectral Data<<<<<", temp_)
        nlines = grep(">>>>>End Processed Spectral Data<<<<<", temp_) - skip - 2

        # actualy reading the data
        data_ = read.csv(file_, sep = "\t", skip = skip, nrows = nlines, dec = ",")

        # create a .png


        png(sub("\\.txt", "\\.png", file_), width = 6, height = 4, units = "in", res = 600)
        par("mar" = c(4, 4, 1, 1))
        par("oma" = c(0, 0, 0, 0)+.2)

        # plot the diagramm
        plot(data_,
                xlab = expression(lambda * " [nm]"),
                ylab = expression(Intensity * " [arb. unit]"),
                type = "l",
                lwd = .5
                )

        # make it more pretty
        abline(h = axTicks(2), v = axTicks(1), lwd = .25, col = "#88888888")
        dev.off()
        cat("\npng saved", sep ="")
        flush.console()
}

for(file_ in files[grepl("Jod2", files)]){
        cat("\nstarting ", basename(file_), sep ="")
        flush.console()
        # read file to get header and relevant linenumbers
        temp_ = readLines(file_)
        skip = grep(">>>>>Begin Processed Spectral Data<<<<<", temp_)
        nlines = grep(">>>>>End Processed Spectral Data<<<<<", temp_) - skip - 2

        # actualy reading the data
        data_ = read.csv(file_, sep = "\t", skip = skip, nrows = nlines, dec = ",")

        # create a .png


        png(sub("\\.txt", "\\.png", file_), width = 6, height = 4, units = "in", res = 600)
        par("mar" = c(4, 4, 1, 1))
        par("oma" = c(0, 0, 0, 0)+.2)

        # plot the diagramm
        plot(data_,
                xlab = expression(lambda * " [nm]"),
                ylab = expression(Intensity * " [arb. unit]"),
                type = "l",
                lwd = .5
                )

        # make it more pretty
        abline(h = axTicks(2), v = axTicks(1), lwd = .25, col = "#88888888")
        dev.off()
        cat("\npng saved", sep ="")
        flush.console()


        # Auswertung
        rng_  = c(495, 600)
        lambd_  = 545.8    # Known peak
        lambd_id = 25    # peak number 25
        lower_end = 505

        x  = data_[[1]]
        y  = data_[[2]]

        rng  = intersect(which(x >= min(rng_)), which(x <= max(rng_)))

        x  = x[rng]
        y  = y[rng]

        ddy  = diff(sign(diff(y)))
        mins = which(ddy == 2)+1
        mins = mins[which(x[mins] > lower_end)]
        peak0 = which(abs(x[mins]-lambd_) == min(abs(x[mins]-lambd_)))
        s_lamd = abs(round(x[mins[peak0]] - lambd_, 1))
        peaks = mins[c(1:(peak0), peak0 + c(2,5))]

        no_peaks= c(116)
        peaks  = peaks[!(peaks %in% no_peaks)]


        png(sub("\\.txt", "\\_zoom.png", file_), width = 6, height = 4, units = "in", res = 600)
        par("mar" = c(4, 4, 1, 1))
        par("oma" = c(0, 0, 0, 0)+.2)
        plot(
                x,
                y,
                xlim = range(x[peaks]) + c(-15, 15),
                xlab = expression(lambda * " [nm]"),
                ylab = expression(Intensity * " [arb. unit]"),
                type = "l",
                lwd = .5
        )
        abline(v = lambd_, col = "#00B00088")
        # points(x[mins[peak0]], y[mins[peak0]], pch = 20, col = "green", cex = 1.5)

        real_peaks = c(mins[peak0])
        sigmas  = c()
        ssigmas  = c()
        As   = c()
        Bs   = c()
        Cs   = c()
        mus   = c()
        for(peak in peaks){
                test_rng = (peak-3):(peak+3)
                min_ = which(y[test_rng] == min(y[test_rng]))

                if(min_ == 4){
                        real_peaks = c(real_peaks, peak)
                        dat = data.frame(x[test_rng], y[test_rng])
                        names(dat) = c("x", "y")
                        fit_g = try(
                                nls(
                                        y ~ A + B * x + C * exp(-(x-mu)^2 / (2 * sigma^2)),
                                        data = dat,
                    start = list(
                        A = max(dat$y),
                        B = 0,
                        C = diff(range(dat$y)),
                        mu = mean(dat$x),
                        sigma = abs(diff(range(dat$x))/4)
                    ),
                    control = list(maxiter = 500000)

                ),
                silent = TRUE
            )
            if(grepl("Error", fit_g[[1]])){
                cat("\nfit not possible:\t", peak, sep = "")
                flush.console()
                points(x[peak], y[peak], col = "red", pch = 4)
                sigmas  = c(sigmas, -1)
                ssigmas  = c(ssigmas, -1)
            } else {
                A  = summary(fit_g)[[10]][1]
                B  = summary(fit_g)[[10]][2]
                C  = summary(fit_g)[[10]][3]
                mu  = summary(fit_g)[[10]][4]
                sigma = summary(fit_g)[[10]][5]/2
                ssigma = summary(fit_g)[[10]][10]/2

                sigmas  = c(sigmas, sigma)
                ssigmas  = c(ssigmas, ssigma)
                As   = c(As, A)
                Bs   = c(Bs, B)
                Cs   = c(Cs, C)
                mus   = c(mus, mu)

                curve(A + B * x + C * exp(-(x-mu)^2 / (2 * sigma^2)), from = x[min(test_rng)], to = x[max(test_rng)], col = "blue", add = TRUE)

            }

        } else{ # NOT A PEAK!!!!
            cat("\nnot a peak:\t", x[peak], sep = "")
            flush.console()
        }
    }
    j = which(sigmas != -1)
    for(i in which(sigmas == -1)){
        idx = j[which(abs(j - i) == min(abs(j - i)))]
        print(idx)
         sigmas[i] = mean(sigmas[idx])
```

```r
        ssigmas[i] = mean(ssigmas[idx])
    }
    print(weighted.mean(abs(sigmas), ssigmas^-2))

    p = unique(x[sort(real_peaks)])
    pid = which(p == x[mins[peak0]])
    id = lambd_id + pid - 1:(length(p)-1) + .5
    p = p * 10^-9
    a = rank(p)
    sigmas = sigmas[a]

    nu = 1 / p / 100
    s_nu = sigmas * 10^-9 / 100 / p^2

    dnu = abs(diff(nu))
    s_dnu = sapply(1:(length(s_nu)-1), function(x)(sqrt((s_nu[x]^2)
    + (s_nu[x+1]^2) )))

    cat("\n")
    flush.console()
    # points(x[real_peaks], y[real_peaks], col = "blue", pch = 20)
    df = data.frame(id, dnu, s_dnu)

    write.csv(df, file = sub("\\.txt", "\\ids.csv", file_), row.names = FALSE, quote = FALSE)
    abline(h = axTicks(2), v = axTicks(1), lwd = .25, col = "#88888888")
    dev.off()


    png(sub("\\.txt", "\\_zoom2.png", file_), width = 6, height = 4, units = "in", res = 600)
    par("mar" = c(4, 4, 1, 1))
    par("oma" = c(0, 0, 0, 0)+.2)
    rng = intersect(which(data_[[1]] >= x[peaks[1]] - 25 ), which(data_[[1]] <= x[peaks[1]] + 25))
    plot(
        data_[[1]][rng],
        data_[[2]][rng],
        xlim = c(490, 510),
        ylim = c(min(data_[[2]][rng]), 20000),
        xlab = expression(lambda * " [nm]"),
        ylab = expression(Intensity * " [arb. unit]"),
        type = "l",
        lwd = .5
    )

    points(x[peaks], y[peaks], pch = 4, col = "#0000B088")
    arrows(499, 15000, 503, 15000, code = 3, angle = 90, length = 0.0175, col = "#B00000")
    lines(c(501, 501), c(0, 15000), col = "#B00000", lwd = .5)
    # lambda = 501nm
    # s_lambda = 2nm

    G = 1 / (501e-9) / 100
    s_G = (2 / 501) * G
    cat("\nWellenzahl des letzen Uebergangs:\n\t", G, "\n\t+-", s_G, "\n", sep = "")
    text(501, 15000, bquote(( 501 %+-% 2 ) * nm), pos = 3)

    abline(h = axTicks(2), v = axTicks(1), lwd = .25, col = "#88888888")
    dev.off()



    # fit toobtain starting values
    fit  = lm(dnu ~ id, weights = s_dnu^-2)

    aa  = coef(fit)[[1]]
    s_aa = summary(fit)$coefficients[3]
    stg  = coef(fit)[[2]]
    s_stg = summary(fit)$coefficients[4]

    id0  = -aa/stg

    # obtaining omegas
    we = - stg / 2 + aa
    s_we = sqrt(s_stg^2 / 4 + s_aa^2)

    wex = -stg / 2
    s_wex = s_stg / 2

    # Gammelfit
    cat("\nstarting Gammelfit")
    flush.console()
    fit_gammel = nls(
            dnu ~ we - wex * (2 * (id-.5) + 2) + wey * (3 * (id-.5)^2 + 6 * (id-.5) * 13 / 4),
            dat = df[1:2],
            start = list(
                    we = - stg / 2 + aa,
                    wex = - stg / 2,
                    wey = 0
            ),
            weights = s_dnu^-2
    )
    we  = summary(fit_gammel)$coefficients[1]
    s_we = summary(fit_gammel)$coefficients[4]
    wex  = summary(fit_gammel)$coefficients[2]
    s_wex = summary(fit_gammel)$coefficients[5]
    wey  = summary(fit_gammel)$coefficients[3]
    s_wey = summary(fit_gammel)$coefficients[6]

    gammel_dnu2_ = function(id){
        return(
            we - wex * (2 * (id-.5) + 2) + wey * (3 * (id-.5)^2 + 6 * (id-.5) * 13 / 4)
            )
    }
    chi2 = sum((gammel_dnu2_(id) - dnu)^2*s_dnu^-2)/ length(s_dnu)
    print(chi2)

    png(sub("\\.txt", "\\_spooner_gammel.png", file_), width = 6, height = 4, units = "in", res = 600)
    par("mar" = c(4, 4.5, 1, 1))
    par("oma" = c(0, 0, 0, 0))

    plot(df$id, df$dnu, ylab = expression(Delta * G * " [" * cm^-1* "]"), xlab = expression(tilde(nu) == nu + frac(1, 2)), pch = 4, ylim = range(df$dnu + df$s_dnu, df$dnu - df$s_dnu, 0))

    arrows(df$id, df$dnu + df$s_dnu, df$id, df$dnu - df$s_dnu, code = 3, angle = 90, length = 0.0175)

    curve(gammel_dnu2_, add = TRUE, col = "red", from = 0 , to = 100)

      upperend = (we - 2 * wex ) / (2 * wex)
    s_upperend = sqrt((s_we / 2 / wex)^2 + (we * s_wex/ 2 / (wex)^2)^2)

      D_nu  = we * upperend / 2
    s_D_nu  = D_nu * sqrt((s_we/we)^2+ (s_wex/wex)^2)

      D_nuE = D_nu * h * c_ * 100 / q
    s_D_nuE = s_D_nu * h * c_ * 100 / q


    abline(h = axTicks(2), v = axTicks(1), lwd = .25, col = "#88888888")
    legend("bottomleft",
        c(
            as.expression(bquote(Delta * G(tilde(nu))==omega[e] - omega[e]*x[e]*(2 * nu + 2) + omega[e]*y[e]*(3 * nu^2+ 6*nu*13/4))),

            as.expression(bquote(omega[e]: (.(round(we, 2)) %+-% .(round(s_we, 2)))*cm^-1)),
            as.expression(bquote(omega[e]*x[e]: (.(round(wex, 3)) %+-% .(round(s_wex, 3)))*cm^-1)),
            as.expression(bquote(omega[e]*y[e]: (.(round(wey, 3)) %+-% .(round(s_wey, 3)))*cm^-1)),
                as.expression(bquote(integral(Delta * G(tilde(nu))*d *nu, 0, nu[diss]) == (.(round(D_nu, 0))%+-% .(round(s_D_nu, 0)))* cm^-1)),
                as.expression(bquote(Chi[red]^2: .(chi2)))
            ),
            lty = c(1, 0, 0, 0, 0, 0),
            col = c("red", "black", "black", "black", "black", "black"),
            cex = .6, bg = "#FFFFFF88"
    )

    dev.off()
    cat("Gammelfit finisched", sep = "")
    cat("\n\nTermdifferenz:\n(", D_nuE, "\t+-", s_D_nuE, ")eV", sep = "")
    flush.console()




    # actual Fit
    fit2 = nls(
            dnu ~ we - wex * (2 * (id-.5) + 2),
            dat = df[1:2],
            start = list(
                we = - stg / 2 + aa,
                wex = - stg / 2
            ),
            weights = s_dnu^-2
    )
    we  = summary(fit2)$coefficients[1]
    s_we = summary(fit2)$coefficients[3]
    wex  = summary(fit2)$coefficients[2]
    s_wex = summary(fit2)$coefficients[4]

    dnu2_ = function(id){
            return(
                we - wex * (2 * (id-.5) + 2)
                )
    }
    s_dnu2_ = function(id){
            return(
                abs(s_we - s_wex * (2 * (id-.5) + 2))
                )
    }
    chi2 = sum((dnu2_(id) - dnu)^2*s_dnu^-2)/ length(s_dnu)
    print(chi2)

    png(sub("\\.txt", "\\_spooner.png", file_), width = 6, height = 4, units = "in", res = 600)
```

```
        par("mar" = c(4, 4.5, 1, 1))
        par("oma" = c(0, 0, 0, 0))

        plot(df$id, df$dnu, ylab = expression(Delta * G * " [" * cm^-1* "]"), xlab = expression(tilde(nu) == nu + frac(1, 2)), pch = 4, ylim = range(df$dnu + df$s_dnu, df$dnu - df$s_dnu, 0))

        arrows(df$id, df$dnu + df$s_dnu, df$id, df$dnu - df$s_dnu, code = 3, angle = 90, length = 0.0175)

        curve(dnu2_, add = TRUE, col = "red", from = 0 , to = 100)

          upperend = round(we/(2* wex) -1)
        s_upperend = sqrt((s_we / 2 / wex)^2 + (we * s_wex/ 2 / (wex)^2)^2)

          D_nu  = upperend * .5 * we
        s_D_nu  = D_nu * sqrt((s_we/we)^2+ (s_upperend/upperend)^2)


          D_nuE = D_nu * h * c_ * 100 / q
        s_D_nuE = s_D_nu * h * c_ * 100 / q


        abline(h = axTicks(2), v = axTicks(1), lwd = .25, col = "#88888888")
        legend("bottomleft",
                c(
                        as.expression(bquote(Delta * G(tilde(nu))==omega[e] - omega[e] *x[e]*(2 * nu + 2))),

                        as.expression(bquote(omega[e]: (.(round(we, 2)) %+-% .(round(s_we, 2)))*cm^-1)),
                        as.expression(bquote(omega[e]*x[e]: (.(round(wex, 3)) %+-% .(round(s_wex, 3)))*cm^-1)),

                        as.expression(bquote(integral(Delta * G(tilde(nu))*d *nu, 0, nu[diss]) == (.(round(D_nu, 0))%+-% .(round(s_D_nu, 0)))*cm^-1)),
                        as.expression(bquote(Chi[red]^2: .(chi2)))
                ),
                lty = c(1, 0, 0, 0, 0, 0),
                col = c("red", "black", "black", "black", "black", "black"),
                cex = .6, bg = "#FFFFFF88"
        )

        cat("\n\nTermdifferenz:\n(", D_nuE, "\t+-", s_D_nuE, ")eV", sep = "")
        flush.console()

        # Morsepotential
          D_e = we^2 / (4 * wex) # [cm^-1]
```

```
s_D_e = sqrt(4 * (s_we/we)^2 + (s_wex/wex)^2 ) * D_e

  D_eE = D_e * h * c_ * 100 / q
s_D_eE = s_D_e * h * c_ * 100 / q
cat("\nMorespotetial:\n(", D_eE, "\t+-", s_D_eE, ")eV\n", sep = "")

flush.console()
dev.off()

# Anregungsenergie
# sigma00 = G0 = G'(nu = n) - n * (Delta G'(.5) + Delta G'(nu + .5) / 2)
       DGhalf = we - wex * (2 * 0 + 2)
       n  = (tail(id, 1)-.5)
         G0 = tail(nu, 1) - n * (DGhalf + (we - wex * (2 * (n+.5) + 2))) / 2
       s_G0 = tail(nu, 1) - n * (DGhalf + (we - wex * (2 * (n+.5) + 2))) / 2

       G0l  = 1 / (G0 * 100 * 10^-9)#nm for comparison with the laser

         G0E = G0 * h * c_ * 100 / q #eV
       s_G0E = s_G0 * h * c_ * 100 / q #eV


# Dissociation energy
# E_diss = simga00 - G(0) + De
       we0  = 214.5200 #cm^-1 literatur
       wex0 = .6079 #cm^-1 literatur
       G00  = we0 - wex0 * (2 * 0 + 2)
       s_G00 = 0 # no error in literature


       G00E = G00 * h * c_ * 100 / q
       s_G00E = s_G00 * h * c_ * 100 / q



       E_diss  = G0E - G00E + D_nuE
       # s_E_diss = s_G0E - s_G00E + s_D_nuE




# Plot morse potential
mu = 1.05e-25
a = sqrt(wex * 100 * 4 * pi * c_ * mu / hbar) * 10^-10  # in Angstrom^-1

s_a = a / (2 * wex) * s_wex # TIM



```

```
        re = 3.02 # in Angstrom

        morse1 = function(r){
                return(D_nuE * (1 - exp(-a * (r - re)))^2)
        }
        morse2 = function(r){
                return(D_eE * (1 - exp(-a * (r - re)))^2)
        }

        png(sub("\\.txt", "\\morse.png", file_), width = 6, height = 4, units = "in"
            , res = 600)
        par("mar" = c(4, 4, 1, 1))
        par("oma" = c(0, 0, 0, 0)+.2)

        curve(
                morse1,
                from = 2.6,
                to = 6,
                xlab = expression(R ~ "["* ring(A)*"]" ),
                ylab = expression(E ~ "[eV]" ),
                col = "#00B000"
        )
        curve(
                morse2,
                xlab = expression(R ~ "["* ring(A)*"]" ),
                ylab = expression(E ~ "[eV]" ),
                col = "#B00000",
                add = TRUE
        )
        abline(h = axTicks(2), v = axTicks(1), lwd = .25, col = "#88888888")
        legend(
                "topright",
                c(
                        expression(D[e["sum"]]),
                        expression(D[e[Mo]])
                ),
                col = c("#00B000", "#B00000"),
                lty = c(1, 1)
        )

        dev.off()
}
```

5.2.2 Calibration

```
graphics.off()
path = dirname(dirname(sys.frame(1)$ofile))

```

```
files = dir(paste(path, "Emission", "Kalibration", sep = "/"), full.names = TRUE, pattern = ".dat")

# literature values
lengths = c(400, 600)
soll_max= c(404.66, 435.83, 546.07, 576.96, 579.07)

# find maxima above value
trigger = 10


for(file_ in files){
        cat("\nstarting ", basename(file_), sep ="")
        # read file to get header and relevant linenumbers
        temp_ = readLines(file_)

        # actualy reading the data
        data_ = read.csv(file_, sep = "\t", dec = ",")
        x  = data_[[1]]
        y  = data_[[2]]

        # find peaks
        m = which(y > trigger)
        m_ = c(1, which(diff(m) > 2), length(m))

        maxs = c()
        #modifying data
        x2 = lengths[1] + diff(range(lengths))/diff(range(x)) * x

        for(i in 2:6){
                rng = m[(m_[i-1]):(m_[i])]
                max_= x2[rng[which(y[rng] == max(y[rng]))]]
                maxs = c(maxs, max_)
                print(max_)
        }


        # create a .png
        png(sub("\\.dat", "\\.png", file_), width = 6, height = 4, units = "in", res = 600)
        par("mar" = c(4, 4, 1, 1))
        par("oma" = c(0, 0, 0, 0)+.1)

        # plot the diagramm
        plot(
                x2,
                y,
                xlab = expression(lambda * " [nm]"),
                ylab = expression(Intensity * " [x*V]"),
                type = "l",
                lwd = .25
                )

        # make it more pretty
        abline(h = axTicks(2), v = axTicks(1), lwd = .125, col = "#88888888")

        # add literature values
        abline(v = soll_max, lty = 2, lwd = .5, col = "#00B000")
        abline(v = maxs, lty = 2, lwd = .5, col = "#880000")

        # text(400, 100, "Lit.", col = "#00B000", pos = 4)
        # text(425, 100, "Mess.", col = "#880000", pos = 4)
        # text(450, 100, expression(Delta * lambda), col = "#000088", pos = 4)
        # for(i in 1:5){
                # text(400, 100 - 7.5*i, soll_max[i], col = "#00B000", pos = 4)
                # text(425, 100 - 7.5*i, round(maxs[i], 2), col = "#880000", pos = 4)
                # text(450, 100 - 7.5*i, round(maxs[i]-soll_max[i], 2), col =
                    "#000088", pos = 4)
        # }
        # text(423, 100 - 7.5*6, "mean:", col = "#000088", pos = 4)
        # text(450, 100 - 7.5*6, round(mean(maxs-soll_max), 2), col = "#000088", pos
             = 4)


        dev.off()
        cat("\npng saved", sep ="")
}
```

5.2.3 Emission

```
graphics.off()
path = dirname(dirname(sys.frame(1)$ofile))

# some constants
h  = 6.6026e-34
hbar = h / 2 / pi
c_  = 3e8
q  = 1.602177e-19

# a function
glaetten = function(x, n){
        temp = c()
        for(i in 1:(length(x)-n)){
                temp = c(temp, mean(x[i:(i+n)]))
        }
        return(temp)
}
wep  = 125.273
```

```
wexp = .70163
Te_lit  = 15770.59
wepp = 214.51886
wexpp = .60738

nups = 1:9
nupps = 4:9
DeltaG = matrix(0, length(nups), length(nupps))
i = 0
for(nup in nups){

       i = i+1
       j = 0
       for(nupp in nupps){
              j = j + 1
              temp = wep * (nup +.5) - wepp * (nupp + .5) - wexp * (nup + .5)^2 +
                  wexpp * (nupp + .5)^2 + Te_lit
              DeltaG[i, j] = temp
       }
}

# DeltaG = DeltaG[c(7, 31), ]
# make a tex file from this table
tex_file = paste(path, "table.tex", sep = "/")

 paste("$\\nu’$", paste("$\\nu’ = ", nupps, "$", collapse = " & "), colapse = " & ",
     sep = " & ")
cat("\\begin{table}[H]\n\t\\begin{center}", file = tex_file)
cat("\n\t\t\\begin{tabular}{ccccc}", file = tex_file, append = TRUE)
cat("\n\t\t\t", header, "\\\\\\hline", file = tex_file, append = TRUE)

for(i in 1:(length(DeltaG[, 1])-1)){
       temp = paste(round(DeltaG[i, ], 0), collapse = " & ")
       cat("\n\t\t\t", nups[i], " & ", temp, file = tex_file, "\\\\", append = TRUE
           , sep ="")

}
i = length(DeltaG[, 1])
temp = paste(round(DeltaG[i, ], 0), collapse = " & ")
cat("\n\t\t\t", nups[i], " & ", temp, file = tex_file, append = TRUE, sep ="")

cat("\n\t\t\\end{tabular}", file = tex_file, append = TRUE)
cat("\n\t\t\\caption{$\\Delta G$}", file = tex_file, append = TRUE)
cat("\n\t\t\\label{tab:deltaG}", file = tex_file, append = TRUE)
cat("\n\t\\end{center}\n\\end{table}", file = tex_file, append = TRUE)


files = dir(paste(path, "Emission", "Beste_Messungen_Jetztwirklich", sep = "/"),
   full.names = TRUE, pattern = ".dat")
```

```

# x-Axis stuff
offset = 0.8439639 # how many nanometers the peak is messured too high, obtained by calibration

lengths_all = list("6400" = c(640, 800), "Laserpeak" = c(632, 635.05))
FCs = c(5e-5, 1.1e-4, 1.7e-4, 1.9e-4, 1.7e-4)


for(file_ in files[1]){
        cat("\nstarting ", basename(file_), sep ="")
        # read file to get header and relevant linenumbers
        temp_ = readLines(file_)

        # actualy reading the data
        data_ = read.csv(file_, sep = "\t", dec = ",")
        x  = data_[[1]]
        y  = data_[[2]]

        lengths = lengths_all[[sub("\\.dat", "", basename(file_))]]
        x2 = lengths[1] - offset + diff(range(lengths))/diff(range(x)) * x

        # create a .png


        png(sub("\\.dat", "\\.png", file_), width = 6, height = 4, units = "in", res = 600)
        par("mar" = c(4, 4, 1, 1))
        par("oma" = c(0, 0, 0, 0))

        # smooth the data
        glaet = c("Laserpeak.dat")
        if(basename(file_) %in% glaet){
                x2 = glaetten(x2, 5)
                y = glaetten(y, 5)
                ylab = expression(Intensity * " [arb. unit]")
        } else{
                y = y / max(y)
                ylab = expression(relatiove~Intensity * " [-]")
        }

        # plot the diagramm
        plot(
                x2,
                y,
                xlab = expression(lambda * " [nm]"),
                ylab = ylab,
                type = "l",
                lwd = .5,
                        col = "#585858"
                        )

        # make it more pretty
        abline(h = axTicks(2), v = axTicks(1), lwd = .125, col = "#88888888")


        if(!grepl("Laser", basename(file_))){
                # Do funny stuff with the data
                rnglng = 200# "rangelength"
                maxima = c()
                s_maxima = c()


                for(i in 0:7500){

                        test_rng = max(c(i - rnglng, 1)):min(c(i + rnglng, length(y)))
                        max_i  = which(max(y[test_rng]) == y[test_rng])

                        if(max_i == min(i, rnglng+1)){

                                print(i)
                                points(x2[i], y[i], col = "red", cex = 3)

                                dat = data.frame(x2[test_rng], y[test_rng])
                                names(dat) = c("x", "y")

                                fit = nls(
                                        y ~ A + B * x + C * exp(-(x - D)^2 / (sigma^2 * 2)),
                                        dat = dat,
                                        start = list(
                                                A = mean(c(head(dat$y, 25), tail(dat$y, 25))),
                                                B = 0,
                                                C = max(dat$y) - mean(c(head(dat$y, 25), tail(dat$y, 25))),
                                                D = x2[i],
                                                sigma = 1
                                        )
                                )

                                s_i  = 2.3548 * summary(fit)$coefficients[5]

                                  maxima = c( maxima, i)
                                s_maxima = c(s_maxima, s_i)
                        }
                }

        E_i  = 1 / (x2[maxima] * 10^-9) / 100
        s_E_i = (s_maxima * 10^-9) / (x2[maxima] * 10^-9)^2 / 50

        temps = c()
        for(i in 1:length(E_i)){
            DE = mapply(function(x)return(abs(E_i[i] - x)), DeltaG)
            temp = which(DE == min(DE))
            temps = c(temps, temp)
        }
        nupp_ = floor(temps / length(DeltaG[, 1]))+1
        nup_ = temps %% length(DeltaG[, 1])


        text(x2[9000], 62, expression(lambda * " [nm]"), pos = 2, cex = .75)
        text(x2[10500], 62, expression("norm."), pos = 4, cex = .75)
        text(x2[9500], 64, expression(Intensity * " [arb. unit]"), pos = 4, cex = .75)

        lines(x2[c(7000, 13000)], c(61, 61))
        for(i in 1:length(maxima)){
            print(i)
            text(x2[9000], 62-(2*i), round(x2[maxima[i]], 2), pos = 2, cex = .75)
            text(x2[9000], 62-(2*i), round( y[maxima[i]], 2), pos = 4, cex = .75)
            text(x2[10500], 62-(2*i), round(y[maxima[i]] / max(y[maxima]), 2), pos = 4, cex = .75)
            text(x2[12000], 62-(2*i), round(E_i[i], 0), pos = 4, cex = .75)
            text(x2[13500], 62-(2*i), round(s_E_i[i], 0), pos = 4, cex = .75)
        }




    } else {
        max_laser = x2[which(y == max(y))]

        dat = data.frame(x2, y)
        names(dat) = c("x", "y")

        fit = nls(
            y ~ A + B * x + C * exp(-(x - D)^2 / (sigma^2 * 2)),
            dat = dat,
            start = list(
                A = mean(c(head(dat$y, 25), tail(dat$y, 25))),
                B = 0,
                                C = max(dat$y) - mean(c(head(dat$y, 25), tail(dat$y, 25))),
                                D = max_laser,
                                sigma = 1
                        )
                )
                A  = summary(fit)$coefficients[1]
                B  = summary(fit)$coefficients[2]
                C  = summary(fit)$coefficients[3]
                D  = summary(fit)$coefficients[4]
                sigma = summary(fit)$coefficients[5]

                laser = function(x){
                        return(A + B * x + C * exp(-(x - D)^2 / (sigma^2 * 2)))
                }
                # curve(laser, add = TRUE, col = "#B0000088")

                s_max_laser  = 2.3548 * summary(fit)$coefficients[5]
                y_ = A + max_laser * B

                # abline(v = max_laser, col = "#B00000")
                # arrows(max_laser - s_max_laser/2, y_, max_laser + s_max_laser/2, y_, code = 3, angle = 90, length = 0.0175, lwd = .5, col = "#B00000")
                # legend("topright", as.expression(bquote(lambda[max]: (.(round(max_laser, 1))%+-% .(round(s_max_laser, 1))) *nm)), bg = "white")
        }

        dev.off()
        cat("\npng saved\n", sep ="")
}
```

6 List of Figures

7 List of Tables

8 Bibliography

[1] Meyer M., *Spektroskopie am Jod-Molekül*, Experiment Instructions, University Freiburg, October 2014

[2] Meyer M., *Verbesserung des Versuchs Spektroskopie am J2- Molekül des Fortgeschrittenen Praktikums*, Teacher's Thesis, University Freiburg, October 2014

[3] Bitsch, K., *Aufbau einer Apparatur fürs Fortgeschrittenen-Praktikum: "Untersuchung der Schwingungsstruktur des Übergangs beim I_2-Molekül"*, Teacher's Thesis, University Freiburg, February 1977

[4] Atkins P. W., Friedman R.S., *Molecular Quantum Mechanics*, Third Edition, Oxford University Press, 1997

[5] Seinfeld, J.I, Zare, N, Jones L., *Spectroscopy Constants and Vibrational Assignment for the $B\ ^3\Pi^+_{0u}$-State of Iodine*, The Journal of Chemical Physics, Volume 42, Number 1

[6] Rank D. H., Rao B. S., *Molecular Constants of the Ground State of I_2*, Journal of Molecular Spectroscopy 13, 34-42 (1964)

[7] Lewis, E.L., Palmer C. W. P ., Cruickshank J. L. , *Iodine molecular constsnts from absorption and laser fluorescence*, Am. J. Phys, Vol 62, April 1994